DOCUMENT RESUME

ED 293 885 TM 011 481

AUTHOR O'Neil, Harold F., Jr.; Baker, Eva L.
TITLE Issues in Intelligent Computer-Assisted Instruction. Testing Study Group: The Impact of Advances in Artificial Intelligence on Test Development.
INSTITUTION California Univ., Los Angeles. Center for the Study of Evaluation.
SPONS AGENCY Air Force Human Resources Lab., Brooks AFB, Texas.; Army Research Inst. for the Behavioral and Social Sciences, Alexandria, Va.; National Aeronautics and Space Administration, Moffet Field, Calif.; Naval Training Systems Center, Orlando, FL.; Office of Educational Research and Improvement (ED), Washington, DC.
PUB DATE Sep 87
GRANT OERI-G086-0003
NOTE 43p.
PUB TYPE Information Analyses (070) -- Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Academic Achievement; *Artificial Intelligence; *Computer Assisted Instruction; Computer Software; Evaluation Methods; *Expert Systems; Formative Evaluation; Summative Evaluation; *Test Construction
IDENTIFIERS *Intelligent CAI Systems

ABSTRACT

Among opportunities to advance the state of the art of intelligent computer-assisted instruction (ICAI) are the evaluation of ICAI systems and the use of the underlying technology in ICAI systems to develop tests. Each issue is addressed via its theoretical context, key constructs, appropriate references to the literature, methodological aspects, and concrete examples of the feasibility of resolving the issue. ICAI systems use artificial intelligence and cognitive science to reach a range of subject matters. Several computer programs are discussed. The key components of ICAI systems include a knowledge base, a student model, and instructional techniques for teaching declarative or procedural knowledge. Research that has contributed to the development of ICAI includes research into both formative and summative evaluation, measurement of student achievement outcomes, measurement of individual differences among students, and process measurement and analysis. A list of 75 references is presented. (TJH)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED293885

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

LISA SIEGEL

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Center for Research on Evaluation,
Standards, and Student Testing

Deliverable - September 1987

Testing Study Group: The Impact of Advances in
Artificial Intelligence on Test Development

ISSUES IN INTELLIGENT COMPUTER-ASSISTED INSTRUCTION

Eva L. Baker

Study Director

Grant Number: OERI-G086-0003

Center for the Study of Evaluation
Graduate School of Education
University of California, Los Angeles

TM011481

The research reported herein was supported in part by the U.S. Department of Education/Office of Educational Research and Improvement, Air Force Human Resources Laboratory, Army Research Institute for the Behavioral and Social Sciences, NASA Jet Propulsion Laboratory, Navy Training Systems Center, and Office of Technology Assesment. However, the views, opinions and/or findings contained in this report are the authors', and should not be construed as an official Department position, policy or decision, unless so designated by other official documentation.

This document will appear as a chapter in *The Computer as Adjunct to the Decision Making Process*, J.C. Conoley, Editor, to be published this year by Lea/Wiley, Hillsdale, New Jersey.

ISSUES IN INTELLIGENT COMPUTER-ASSISTED INSTRUCTION: EVALUATION AND MEASUREMENT

HAROLD F. O'NEIL, JR.

University of Southern California

EVA L. BAKER

Center for Research on Evaluation, Standards, and Student Testing

University of California, Los Angeles

In this chapter we plan to explore two issues in the field of intelligent computer assisted instruction (ICAI) that we feel offer opportunities to advance the state of the art. These issues are evaluation of ICAI systems and the use of the underlying technology in ICAI systems to develop tests. For each issue we will provide a theoretical context, discuss key constructs, provide a brief window to the appropriate literature, suggest methodological solutions and conclude with a concrete example of the feasibility of the solution from our own research.

INTELLIGENT COMPUTER-ASSISTED INSTRUCTION (ICAI)

ICAI is the application of artificial intelligence to computer-assisted instruction. Artificial intelligence, a branch of computer science, is making computers smart in order to (a) make them more useful and (b) understand intelligence (Winston, 1977). Topic areas in artificial intelligence have included natural language processing (Schank, 1980), vision (Winston, 1975), knowledge representation (Woods, 1983), spoken language (Lea, 1980), planning (Hayes-Roth, 1980), and expert systems (Buchanan, 1981). The field of Artificial Intelligence (AI) has matured in both hardware and software. The most commonly used language in the field is LISP (List Processing). A major development in the hardware area is that personal LISP machines are now available at a relatively low cost (20-50K) with the power of prior mainframes. In the software area two advances stand out: (a) programming support environments such as LOOPS (Bobrow & Stefik, 1983) and (b) expert system tools. The application of "expert systems" technology to a host of real-world problems has demonstrated the utility of artificial intelligence techniques in a very dramatic style. Expert system technology is the branch of artificial intelligence at this point most relevant to ICAI.

Expert Systems

Knowledge-based systems or expert systems are a collection of problem-solving computer programs containing both factual and experiential knowledge and data in a particular domain. When the knowledge embodied in the program is a result of a human

expert elicitation, these systems are called expert systems. A typical expert system consists of a knowledge base, a reasoning mechanism popularly called an "inference engine" and a "friendly" user interface. The knowledge base consists of facts, concepts, and numerical data (declarative knowledge), procedures based on experience or rules of thumb (heuristics), and causal or conditional relationships (procedural knowledge). The inference engine searches or reasons with or about the knowledge base to arrive at intermediate conclusions or final results during the course of problem solving. It effectively decides when and what knowledge should be applied, applies the knowledge and determines when an acceptable solution has been found. The inference engine employs several problem-solving strategies in arriving at conclusions. Two of the popular schemes involve: Starting with a good description or desired solution and working backwards to the known facts or current situation (backward chaining); and starting with the current situation or known facts and working toward a goal or desired solution (forward chaining). The user interface may give the user choices (typically menu driven) or allow the user to participate in the control of the process (mixed initiative). The interface allows the user: to describe a problem, input knowledge or data, browse through the knowledge base, pose question, review the reasoning process of the system, intervene as necessary, and control overall system operation. Successful expert systems have been developed in fields as diverse as mineral exploration (Duda & Gaschnig, 1981) and medical diagnosis (Clancy, 1981).

ICAI Systems

ICAI systems use approaches artificial intelligence and cognitive science to teach a range of subject matters. Representative types of subjects include: (a) collection of facts, e.g., South American geography in SCHOLAR (Carbonell & Collins, 1973); (b) complete system models, e.g., a ship propulsion system in STEAMER (Stevens & Steinberg, 1981) and a power supply in SOPHIE (Brown & Burton, 1978); (c) completely described procedural rules, e.g., strategy learning, WEST (Brown, Burton, & de Kleer, 1982), or

arithmetic in BUGGY (Brown, Burton, & Larkin, 1977); (d) partially described procedural rules, e.g., computer programming in PROUST (Johnson & Soloway, 1983); LISP Tutor (Anderson, et al., 1985); rules in ALGEBRA (McArthur, Stasz & Hotta, 1987); diagnosis of infectious diseases in GUIDON (Clancey, 1979) ; and an imperfectly understood complex domain, causes of rainfall in WHY (Stevens, Collins, & Goldin, 1978). Excellent reviews by Barr and Feigenbaum (1982) and Wenger (1987) document many of these ICAI systems. Representative research in ICAI is described by O'Neil, Anderson and Freeman (1986).

Although suggestive evidence has been provided by Anderson et al. (1985), few of these ICAI projects have been evaluated in any rigorous fashion. In a sense they have all been toy systems for research and demonstration. Yet, they have nonetheless raised a good deal of excitement and enthusiasm about their likelihood of being effective instructional environments.

With respect to cognitive science, progress has been made in the following areas: identification and analysis of misconceptions or "bugs" (Clement, Lockhead, & Soloway, 1980), the use of learning strategies (O'Neil & Spielberger, 1979; Weinstein & Mayer, 1986), expert versus novice distinction (Chi, Glaser, & Rees, 1982), the role of mental models in learning (Kieras & Bovair, 1983), and the role of self-explanations in problem solving (Chi, Bassok, Lewis, Reimann, & Glaser, 1987).

The key components of an ICAI system consist of a knowledge base: that is, (a) what the student is to learn; (b) a student model, either where the student is now with respect to subject matter or how student characteristics interact with subject matter; and (c) a tutor, that is, instructional techniques for teaching the declarative or procedural knowledge. These components are described in more detail by Fletcher (1985).

Knowledge Base. This is the "expert" part of the system. Ideally, this component would represent the relevant knowledge domain. In effect, it must contain the knowledge and understanding of a subject matter expert. It must be able to generate problem solutions

from situations never before encountered and not anticipated by the training system designers. It must be able to infer the true state of the system from incomplete and/or inaccurate measurements. It must be able to solve problems based on this knowledge.

Student Model. This component represents the learner. Just as the knowledge base must "understand" the subject matter, so the student model must understand and be able to model the learner. The function of the student model is to assess the student's knowledge state and to make hypotheses about his/her conceptions and reasoning strategies. There are two main approaches to student modeling: (1) the overlay model, in which a model is constructed by comparing the student's performance to the computer-based expert's behavior on the same task. Thus, the student's knowledge state is a subset of an expert's knowledge (Carr & Goldstein, 1977); and (2) the buggy model which represents student's mislearned subskills as variants of the expert's knowledge. Thus, misconceptions are modeled as incorrect procedures (Brown & Burton, 1978). Some systems emphasize a student's knowledge/gaps in his knowledge base. Others emphasize students' misconceptions. Few do both of these very well; however, none of the current ICAI systems represent the role of traditional individual differences (i.e., smart students learn faster than not-so-smart students [Sternberg, 1982]).

Tutor. This component represents the teacher and must be able to apply the appropriate instructional tactics at the appropriate times. This capability implies the presence of both a large repertoire of instructional tactics and a strategic understanding of how best to use them. It should model the desirable properties of a human tutor. Figure 1 presents some of these properties. In general, the tutor must know what to say to the learner and when to say it. In addition, it must know how to take the learner from one stage of skill to another and how to help the learner, given his or her current state of knowledge.

Insert Figure 1 about here

However, little of instructional design considerations (e.g., Ellis, Wulfeck & Fredericks, 1979; Markle, 1967; Merrill & Tennyson, 1977; O'Neil, 1979; Park, Perez & Seidel, 1987; or Reigeluth, 1987) are reflected in ICAI tutors. Instructional design is concerned with "prescribing optimal methods of instruction to bring about desired changes in student knowledge and skills" or alternatively is viewed as a "linking science . . . a body of knowledge that prescribes instructional actions to optimize designed instructional outcomes, such as achievement and affect" (Reigeluth, 1983). More recently, there have been several systematic attempts to provide instructional information into the design of ICAI systems. Such attempts include the design of a new ICAI tutor (O'Neil, Slawson, & Baker, 1987) and the design of instructional strategies to improve existing ICAI programs (Baker et al., 1985). However, neither of these efforts systematically evaluated the resulting "improved" ICAI programs. Research in progress by McArthur of the Rand Corporation is addressing this issue in the domain of algebra..

Evaluation

Evaluation is an activity purported to provide an improved basis for decision-making. Among its key elements are the identification of goals, the assessment of process, the collection of information, analysis, and the interpretation of findings. A critical issue in any sort of evaluation is the meaning ascribed to the findings. Meaning derives from the use of measures that are valid for the intervention, from the adequacy of the inferencing processes used to interpret results, and from the utility of the findings for the intended users. These facets of meaning require that the designer/developer as well as funding sources articulate their goals, processes, and potential decision needs so that the evaluation team can provide results that have meaning for interested parties.

Summative Evaluation. The most common model for evaluation is summative (Scriven, 1967) which focuses on overall choices among systems or programs based upon performance levels, time, and cost. In this mode, evaluation is essentially comparative and contrasts the innovation against other options. These comparisons may be against explicit

choices or may be implicit in terms of current practice or ways resources might be spent in the future (opportunity costs).

Summative evaluation asks the question, "Does the intervention work?" In a military or industrial training environment, a common question is, "Has training using X approach been effective?" Implicit in that question is comparison, for the intervention must be judged in comparison to other alternatives, either current practice, or hypothetically, in terms of other ways the resources could be used. A second part of the summative evaluation question is "How much does it cost?" Again, comparisons may be implicit or explicit. Third, summative evaluation develops information related to a third, critical question: "Should we buy it?" Here, the issue is the confidence we have in our data, and the validity of the inferences we draw from such data. We judge the credibility of our cost information case against the validity and credibility of quality data and cost of competing alternatives.

Where summative evaluation is weak is in identifying what to do if a system or intervention is not an immediate, unqualified success. Given that this state is most common for most interventions in early stages of development, comparative, summative type evaluations are usually mis-timed and may create an unduly negative environment for productivity. Furthermore, because summative evaluation is typically not designed to pinpoint weaknesses and to explore potential remedies, it provides almost no help in the development/improvement cycle which characterizes the systematic creation of training interventions.

Formative Evaluation. Evaluation efforts that are instituted at the outset or in the process of an innovation's development typically have different purpose. Formative evaluation (Baker, 1974) seeks to provide information that focuses on the improvement of the innovation and is designed to assist the developer.

Formative evaluation also addresses, from a metaevaluation perspective, the effectiveness of the development procedures used, in order to predict whether the

application of similar approaches will likely have effective and efficient results. In that function, formative evaluation seeks to improve the technology at large, rather than the specific instances addressed one at a time. The approach, formative evaluation, is designed so that its principal outputs are identification of success and failure of segments, components, and details of programs, rather than a simple overall estimate of project success. The approach requires that data be developed to permit the isolation of elements for improvement and, ideally, the generation of remedial options to assure that subsequent revisions have a higher probability of success. Formative evaluation is a method that developed to assist in the development of instructional (training) programs. While the evaluation team maintains "third-party" objectivity, they typically interact with and understand program goals, processes, and constraints at a deeper level than evaluation teams focused exclusively on bottom line assessments of success or failure. Their intent is to assist their client (either funding agency or project staff) to use systematic data collection to promote the improvement of the effort.

Basic literature in formative evaluation was developed by Scriven (1967), Baker and Alkin (1973), Baker (1974), and Baker and Saloutos (1974). Formative evaluation now represents the major focus of evaluation efforts in the public education sector (Baker & Herman, 1985) in the guise of instructional management systems. Multiple models and procedures are common within formative evaluation. An example of one approach to formative evaluation for ICAI is depicted in Figure 2. As is shown in Figure 2, formative evaluation begins with checking whether the design is congruent with specifications and ends with revision which includes new data collection on Steps 3-5. An attempt to use this approach was conducted by Baker et al. (1985).

Insert Figure 2 about here

Tensions in Evaluation. A persistent fact of evaluation is that those evaluated rarely see the value of the process. It is something done to them, a necessary evil, a new chance for failure, often seen as largely irrelevant to their major purpose. This view

generally holds whether it is a person who is evaluated (for selection or credentialling purposes), such as students and teachers at universities or in the public schools, a program evaluated (either as small as a segment or as large as a federal initiative), or a technological innovation. Those who get evaluated are almost always reluctant players.

As persistent a fact, however, is that those in authority have come to believe that evaluation is a useful process. Their belief is fostered in part by actual research studies showing that evaluation findings, when used, improve the state of affairs. But a more likely reason that evaluation has been fastened upon as a useful endeavor resides in the belief that it provides a mechanism for management, or for the appearance of management, by those in charge of resources. Objectivity, accountability, and efficiency are themes underlying this commitment to evaluation.

The tension is obvious between those who must participate and those who push the evaluation process from positions of authority. Evaluation experts have to mediate among these two sets of views, a challenging, if not always pleasant task.

The Evaluability of ICAI Applications. Evaluating an emerging technology presents serious technical as well as practical problems, and the ICAI field incorporates most known or imaginable difficulties. First, much has been claimed by proponents of Artificial Intelligence (AI). The claims have led many sponsors to support projects that they believe intend to produce a fully developed instructional innovation (such as a tutor). In fact, the intention of the designers may not be to create a working, effective tutor, but to work toward this goal and thereby to explore the limits of the computer science field. In this case, the tutor becomes a context for R&D, a constraint under which the designer really seeks to conduct research, i.e., produce new knowledge about AI processes. Such a process makes sense in an emerging field but requires great patience from sponsors.

Because ICAI efforts develop largely in a research rather than in a development context, certain facts characterize them. First, research goals contributing to knowledge and theory building appear to be paramount. Focusing on academically respectable efforts

frequently characterizes emerging, synthetic fields. (See, for instance, the spate of theory building in educational evaluation in the late 60's.) Second, efforts are selectively addressed based on the research predilections (rather than the project development requirements) of any particular set of investigators. Third, there are no real "off-the-shelf-item" components available for easy substitution into the project. Thus, if the researcher invests effort in knowledge representation, his final product may not "work" because of the lagged emphasis in another important component, e.g., a tutor. The foreknowledge of uncertain success to the researcher need not impair the ICAI enthusiasm. Again, rhetoric of the goal of a complete ICAI system is useful. In an emerging field, breakthroughs are anticipated. Secondly, keeping the idea, even as an idea, of a complete future ICAI in the mind of the researcher suggests fruitful paths of exploration.

Thus, the lines between research and application in ICAI are murky and undercut neat categories of R&D processes, such as those identified by Glennan (1968) and Bright (1970) and used as program elements in DoD work (Basic Research [6.1], Exploratory Development [6.2], Advanced Development [6.3], and Engineering Development [6.4]). This reality presents problems for evaluation. Compared to other innovations, the ICAI *what* to be evaluated is less concrete and identifiable, and more like the probabilistic view of where a photon is at any point in time. In addition, the field of ICAI uses multiple metaphors to describe its activity. Figure 3 depicts these multiple metaphors. We believe that each setting requires a different role for the student and, thus, a different evaluation focus.

Insert Figure 3 about here

Secondly, ICAI has evaluability problems partly because of its visibility; the public persona of AI (see national magazines, films, television, trade books) is high profile. In startling contrast, the accessibility to AI processes is limited. To the uninitiated, it is embedded in the recesses of special language (e.g., LISP, PROLOG) and in arcane jargon (modified petri net, overlay models). Coupled with the fact that AI work is conducted in a

relatively few centers by a relatively small number of people, understanding an AI implementation well enough to create sensible options for its assessment is a difficult proposition. These states are compounded by the strongly capitalistic environment in which AI research is conducted. The proprietary nature of much work, either that conducted by large private corporations or by small entrepreneurial enterprises also works to obscure the conceptual and procedural features of the work. AI applications are unlike, therefore, innovations in health, criminal justice, education, industrial training, employment, or transportation because of the lack of mid-level communication about what the innovation actually is. Perhaps AI experts can assist in evaluation, but, understandably, they are more interested in creating something new of their own. All of this is asserted with full knowledge that at least some of these problems characterize any rapidly developing new technology.

The utility of evaluation processes also needs to be judged in terms of what techniques and options are useful, where there is differential confidence in our ability to measure and infer, and which procedures have been used credibly in the last ten years. In addition, we must consider what requirements ICAI evaluation creates and explore new methodology to meet these needs. We have begun to develop such a methodology. Table 1 presents questions we believe that an ICAI evaluation should answer and thus increase the evaluability of ICAI.

Insert Table 1 about here

Distance Between the Evaluator and the Evaluated. One way to think about either formative or summative evaluation techniques is in terms of the distance among those who are conducting the evaluation work, those responsible for the actual day-to-day design and development of the project, and those who are responsible for providing resources to the project. These distances are often represented as the "party" of the evaluation.

First party evaluation is evaluation conducted by the project staff itself. Common examples would be pilot test data conducted for input into the design of a final project. It

has the benefit of intimate connection and understanding of the project. Its problem is lack of distance and detachment. In AI applications, this evaluation work is informal, and relatively infrequently addressed to the issue of overall effectiveness of the intervention. Further, many ICAI projects are conceptualized to advance the state of the art in computer science (a view of the developer). This perspective may conflict with the view of the funder of a project to create an ICAI system with of an instructionally sound tutor.

Second party evaluation involves the assessment of progress or outcomes by the supervising funding agency. IPRs and site visits are examples of second party evaluation. Arbitrary timing, limited agency attention spans, and objectivity are problems here. Further, a real intellectual give and take is difficult when agency personnel control funds.

Third party evaluation is evaluation conducted by an independent group. GAO performs many third party summative evaluations. Independent contractors reporting to state legislatures, school boards or school districts also conduct such evaluation. The benefit of such an approach is the disinterested nature of the investigation, contributing to the credibility of the findings. However, the validity of external evaluation presents some difficulty, and requires, however, that the third party get up to speed in technical issues so that the evaluation methodologies applied are appropriate. The learning required by the evaluation staff represents an additional "overhead" to the project staff and may be perceived as a distraction from their primary effort. This sort of evaluation costs more than the other two.

All types of above evaluation can be done using formative or summative techniques. Third party formative evaluations are rare in general and to our knowledge have only been applied once in ICAI (Baker et al., 1985).

Evaluation Technology. Contrary to popular practice, there is no inherent reason for totally separating formative and summative evaluation efforts. We have mentioned that the approaches differ in purpose and client. They also differ in the types of data appropriate (cost for summative, componential analysis for formative). However, in the

area of performance, they should share some common procedures and criterion measures. In addition, since ICAI shares some common attributes with CAI, evaluation technology appropriate to CAI could be used in ICAI (e.g., Merrill et al., 1986; Alessi & Trollip, 1985). The CAI lesson evaluation techniques in Table 2 present some formative (quality review and pilot testing methods) and some summative techniques (i.e., validation). These activities were adapted from Alessi and Trollip (1985). Information of this sort is a necessary but not sufficient set for ICAI evaluation. What is missing in Table 2 and needs to be developed for ICAI are specific procedures that focus on the unique attributes of ICAI. Table 3 provides a first cut of such attributes. To our knowledge, there are no known techniques to evaluate systematically and instructionally the features in Table 3. However, an interesting approach for the analysis of rapid prototyping is provided by Carroll and Rosson (1984), and Richer (1985) discusses knowledge acquisition techniques.

Insert Table 2 about here.

Insert Table 3 about here

It is not likely that evaluation as it is currently practiced can be transferred directly to an application field such as ICAI. One approach to exploring the merging of existing technologies (ICAI applications with evaluation technology) is to shift points of view in order to determine where reasonable matches exist. Looking first from the evaluation perspective, let us explore where evaluation has some strengths and could make a substantial contribution to ICAI development.

Evaluation's Contribution to ICAI

Research and development in measurement is one of the major productive areas in psychology. Sophisticated models for estimating performance have been developed and come in and out of vogue. Many of these were created to assist in the selection process, to sort those individuals who were better or worse with regard to a particular competency or academic domain. However, these approaches, while venerable, have little to contribute to

the evaluation of programs, either those completed or under continuing development. Most standardized achievement tests were based on this model, and their use to evaluate innovation is not recommended for a variety of technical reasons. These reasons can be summed up on a simple phrase: standardized tests are not sensitive enough to particular curriculum foci; thus, they are unlikely to detect effects present (the false negative problem) and will underestimate effects that exist.

- **Measurement of Student Achievement Outcomes**. However, there are newer approaches to the measurement of human performance which do have implications for the assessment of ICAI interventions designed to improve learner performance. Specifically, the use of domain-referenced achievement testing seems to provide a good match with ICAI approaches. In domain-referenced testing (Hively, Patterson, & Page, 1968; Baker & Herman, 1983; Baker & O'Neil, 1987) one attempts to estimate student performance in a well-specified content domain. The approach is essentially top-down, with parameters for content selection and criteria for judging adequacy of student output specified (albeit successively revised) in advance. Test items are conceived as samples from a universe constrained by the specific parameters. For example, in the area of reading comprehension, parameters would need to be explicated regarding the genre and content to be read, the characteristics of the semantics and syntax, including variety, ambiguity, complexity of sentence patterns, and the presupposed knowledge that the learner would bring into the instructional/testing setting. In addition, the characteristics of the items would be identified, in terms of gross format, i.e., short answer, essay, multiple choice, and in terms of subtler features such as the rules for the construction of wrong answer alternatives, or for the assessment of free responses. Theoretically, such rules permit the generation of a universe of test items which can be matrixed resampled to provide progress and end-of-instruction testing.

The use of such approaches have the added benefit of utility to small numbers of students. They do not depend, as does the selection approach described above, upon

normal (and large) distributions of respondents to derive score meaning. On the other hand, such tests are more demanding to develop, and they depend upon close interaction with the innovation designer to assure that the specifications are adequate. They contrast with the common approach of "tacking on" existing measures (like commercially available standardized tests), an easy enough process but one unlikely to provide information useful for the fair assessment of improvement of a product. Domain-referenced tests derive their power from the goodness of their specifications. Their weakness is their idiosyncrasy; however, the matching of testing procedures to designer's intentions is also their strength.

Because of the attention that ICAI applications devote to representing properly the knowledge domain and determining student understanding in process, the application of improved assessment techniques, particularly those based on domain-referenced testing seems like a good fit.

- **Measurement of Individual Difference**. A second area in measurement that could contribute to the efficient design and assessment of ICAI applications is the measurement of individual differences. Psychology has long invested resources in determining how best to assess constructs along which individuals show persisting differences. For these areas to be useful, such constructs should interact (statistically) with instructional options and desired outcomes of the system under study (Corno & Snow, 1986). Common constructs such as ability and intelligence undoubtedly have relevance for the analysis and implementation of alternative student models and tutoring strategies. Other constructs related to cognitive style preferences, e.g., the need for structure, the need for reflection, the attribution of success and failure, could illuminate design options and results analyses for ICAI applications. Similarly, constructs related to affective states, i.e., state anxiety (Hedl & O'Neil, 1977), could also provide explanations of findings otherwise obscure.

- **Process Measurement and Analysis**. In formative evaluation, much is made of the role of process evaluation, that is, tracking what occurs when, to assure that

inferences about system effectiveness are well placed. Central to this function, however, is deciding, to the extent possible, what data should be collected and which inferences should be drawn from the findings. Technology-based innovations often make two seemingly conflicting classes of errors. One error is collecting everything possible that can be tracked. Student response times, system operation, errors, student requests, etc., can be accumulated ad nauseum. The facts seem to be that rarely do developers attend to this glut of information. They have no strategies for determining how such data should be arranged in priority, nor ways to draw systematic conclusions from findings. By the time the database is assembled, developers are often on to new ideas and prospects; old data, particularly painfully analyzed and interpreted old (to the developer) data, remain only old and often unused. The other error in technology process measurement is when relevant information which could be painlessly accumulated and tabulated on-line is ignored.

The challenge for the evaluator is to help decide what data are likely to be most relevant. Relevance will presuppose a clear overall goal, such as teaching a target group a set of skills. In fact, in the entire gamut of measurement options available, the most significant contributions evaluators may make is clarifying the goals that the designer possesses but has not articulated. Because of the mixture of research and development goals inherent in much ICAI work in education, this is a nontrivial problem. The designers may feel they have all the goals they can tolerate.

- **Generation of Instructional Options**. Formative evaluators can assist ICAI designers to explore different ways in which they can successfully meet their goals. Of particular interest, for example, is the extent to which evaluation can highlight alternatives for the instructional strategies used in the application. In all instructional development, not the least in ICAI-based approaches, the designer fastens early upon a particular strategy. Research findings have suggested that teachers and developers are most reluctant to change the approach they have taken. They will play at the edges rather than rethink their overall method (Baker, 1976). Furthermore, they could easily adapt their basic approach by

adding particular instructional options to their basic plan, assuming that they make their choice informed by prior research. A recent study (Baker, Bradley, Ashbacher, & Feifer, 1985) adopted such an approach and experimentally modified WEST to strengthen its teaching capability. Although largely unsuccessful due to implementation issues, it demonstrated the feasibility of the concept.

Formative Evaluation of ICAI: A Case Study

This section will focus on the Baker et al. (1985) formative evaluation of PROUST as an example of a formative evaluation of ICAI. PROUST (Johnson & Soloway, 1983, 1987) was selected by Baker et al. (1985) as one of the projects to formatively evaluate because its designers communicated serious interest in whether PROUST was instructionally effective with students.

Evaluation Focus. A three-phase evaluation template was designed for use in the project evaluation. The first phase of the evaluation included an attempt to understand the "product" development cycle employed, the ideological orientations of the designers, and their stated intentions. A second phase of analysis involved reviewing the internal characteristics of the ICAI systems from two perspectives: first, the quality of the instructional strategies employed; and second, the quality of the content addressed. A third and major phase of the study was empirical testing of the programs. Here, the intention was to document effects of the program with regard to individual difference variables among learners and with regard to a broadly conceived set of outcome measures including achievement and attitude instruments. An explicit intent was to modify the instructional conditions under which the ICAI system operated and make it more effective. Planned experimental comparisons were one option by which these instructional conditions could be contrasted. Based on these three major phases (theoretical, instructional, and empirical analyses), recommendations for the improvement of this particular project and for the ICAI design and development process in general were to be developed. A wide range of

evaluation techniques were to be included, for instance, both quantitative and qualitative data collection and analyses. This process is a variant of Figure 2.

Evaluation Questions. The evaluation questions guiding the study are presented below. These questions are a variant of Tables 1, 2, and 3. In each of these, information related to the adequacy of the AI components (i.e., knowledge representation, instructional strategy, and student model) are treated as appropriate.

1. What is the underlying theoretical orientation of the system under evaluation? To what extent does the program serve as a model for ICAI?
2. What instructional strategies and principles are incorporated into the program? To what extent does the project exhibit instructional content and features potentially useful to future Army applications?
3. What are the learning outcomes for students? To what extent do learners achieve project goals? Do students with different background characteristics profit differentially from exposure to the project? To what extent does the program create unanticipated outcomes, either positive or negative?

Each of these questions was applied to the PROUST ICAI project.

PROUST: Program Description. The ICAI system entitled PROUST was designed by Johnson and Soloway at Yale University. The system title is a literary allusion: *Remembrances of Bugs Past*, with apologies to the original author.

PROUST is designed to assist novice programmers to use the PASCAL language in their own writing of computer programs. The approach taken is to provide intelligent feedback to beginning students about the quality of their efforts in an attempt to approximate the feedback that a human tutor might provide. In the words of its designers, PROUST is:

"... a tutoring system which helps novice programmers to learn to program."

"... a system which can be said to truly understand (buggy) novice programs," (Johnson & Soloway, 1983).

Thus, PROUST is not a trivial effort. The designers have had to map the cognitive domain of computer programming, with PASCAL as the specific instance. The evaluated implementation (circa 1985) of PROUST permited students to submit their programs in

response to two specific (but intended to be prototypic) programming problems. PROUST takes as its input programs which have passed through the PASCAL compiler and are syntactically correct. In analyzing these programs, PROUST attempts to infer students' intentions and to identify any mistakes (bugs in their software) that occurred in the code (Johnson & Soloway, 1983).

As an example of a functioning ICAI system, PROUST represents only a partial solution for the need to formatively evaluate a complete ICAI system. It contains the knowledge representation in software for the problemspace of the specific PASCAL programming problems. It also contains the diagnostic part of a tutoring component, which analyzes the student program to determine both student intentions and bugs. PROUST then provides feedback about its inferences about students' intentions and how well the student program implements the assumed plans. However, it does not have a robust tutor. Currently (circa 1987) under development is the pedagogical expert, which knows how to interact with and instruct (tutor) students effectively, and contains a student model to cumulatively monitor student progress. Although it has been anticipated that these components would be available for a full test of the ICAI system, schedule constraints restricted our activities to the completed components. The Yale project staff attempted to include an additional level of feedback in the analyzer as a precursor to the full development of the tutor.

Evaluation Approach. As was discussed previously, for the evaluation of PROUST, three sets of questions guided our efforts. The evaluation questions, dimensions of inquiry, measurement method, and data sources guiding the study are presented in Table 4.

Insert Table 4 about here

Because the questions clearly call for a variety of data collection an analysis, ranging from review of documentation, inspection of the program, close observation of outputs from the programs, and student performance and self-report information, the procedures in

the study were complex. Thus, Table 4 summarizes the instrumentation, data collection, and respondents required for aspects of the program under review.

Formative Evaluation Results. The report by Baker et al. (1985) presents the complete description and evaluation of PROUST. There are three major sections of their document: a theoretical analysis of the program, a formative review, and a report of two effectiveness studies conducted with PROUST. As was discussed, the purpose of their evaluation was to provide information relevant to the potential improved effectiveness of the system. For the purposes of this chapter, we will provide a concise summary of their findings. We suggest that their methodological approach and measuring procedures are appropriate for a formative evaluation of ICAI systems in general.

The theoretical orientation of PROUST is a top-down approach based on intentions and plans. Rather than compare the student program to an ideal implementation, PROUST compares it to the plan it believes the student was attempting. PROUST inspects a student's program and attempts to classify the inferred intentions against a set of possibilities based upon prior student approaches. The program's greatest strength is perhaps its ability to deal with alternative goal decompositions. Its weakness is that it does not explicitly ask the student to confirm the plan that the program "thinks" the student is pursuing.

Because PROUST was only a partial ICAI system, irecommendations for improvement focused on two instructional features: type of feedback provided to students and bug analysis. Suggestions for improving feedback were made, especially the content, tone, and learner-control of feedback. Additional recommendations were made for increasing the interactive aspects of PROUST's implementation through verification of student plans, input/output analysis, and student control of timing. In general, Baker et al.'s (1985) study showed few significant findings of use of PROUST related to learning outcomes. However, students were generally positive about using the program. The

designers continue their own evaluation efforts, and Soloway has recently presented workshops (circa 1987) on the topic.

How Can Evaluation Assist ICAI Applications: Some Suggestions

The history of evaluation of ICAI implementations is light reading. For evaluation to work to the mutual benefit of application designers and their resource providers, we suggest the following:

1. The expectation of evaluation should be developed in the minds of the ICAI developers. The description of the instructional effectiveness of applications needs to become part of the socialized ethic, as in science, the expectation of repeatability, verifiability and public reporting is commonplace.

2. Rewards for designers' participation in evaluation are necessary. These must be over and above the intrinsic value of the evaluation information for the designer. Because evaluation is not a common expectation, special benefits must be developed to create cooperation.

3. The credibility of the evaluation team must be seriously addressed. AI experts need to participate in AI and ICAI evaluations. Their participation needs to depend less on frantic persuasion and more on a developed sense of professional responsibility (like reviewing for a journal). If the approach taken is formative, then the designer can receive "help" from friendly reviewers. The goal of evaluation of this sort is to aid in revision rather than to render a judgment.

4. Approaches to evaluation must take account of specific features of ICAI development. Rather than waiting for the completed development, the evaluation team can assist in some decision making related to instruction or utilization. While this sounds easy, it depends upon the view that "outsiders" know psychology or performance measurement in ways that may be useful to ICAI experts. We need to overcome the "not invented here" syndrome.

5. Evaluation needs to be componential and focus on the utility of the piece of software under development. Records of rapid prototyping and redesign need to be integrated into the formative evaluation. It is as useful to record the blind alleys as the successes.

6. Evaluation needs to be responsible and responsive. Objectivity must be preserved, but at the same time, those evaluated must not feel victimized. A reasonably positive example occurred in the formative evaluation of PROUST (Baker, et al., 1985). Among the most interesting phases of that activity was the dialogue following the submission of the draft of the report to Dr. Soloway. Through an interactive process, the evaluation report was strengthened, fuller understanding of the intentions and accomplishments of the project staff were developed, and points of legitimate disagreement were identified. In all cases, the AI expert was able to present (directly quoted) his point of view. The overall outcome was that the fairness of the report was not questioned.

Artificial Intelligence and Test Development

Although AI has a number of branches that may have educational implications (e.g., work in vision to assist the handicapped student), our interest in this section of our chapter will focus on the processes related to the design of expert systems and intelligent computer assisted instruction (ICAI) as they may help to improve test design. We believe that this technology has enormous implications for the creation of rigorous test materials in the future. Expert systems provide an opportunity for specific knowledge domains to be identified, structured, and incorporated into computer software, while efforts in cognitive science have focused on alternative forms of representing such knowledge accurately and completely.

The expertise of "expert" systems sometimes comes from comparing the problem solving approaches of skilled people and attempting to represent them within the computer, thus allowing the computer to perform tasks with equivalent expertise (although often with greater speed and reliability). The techniques to represent knowledge developed for AI

expert systems could potentially be used in the vexing problems of assuring full content representation on tests. Because content of tests (especially those commercially produced) varies enormously in depth, comprehensiveness, and accuracy (Herman & Cabello, 1983; Floden, et al., 1980; Baker & Quellmalz, 1980; Burstein, et al., 1985), using a knowledge representation approach may in itself be a contribution for test development, even without incorporating it as part of a complex, computer-delivered system. Content sampling, and theory in support of it, is an area of continuing weakness in many test development activities, particularly those which are locally based.

Knowledge representation is the core of any ICAI system. It focuses on what is the principal database of interest, which is a knowledge base. Since expert systems combine the idea of knowledge base and representation with the expert's "wisdom," pertinent issues to this area in the testing field are: (1) who are the experts (subject matter specialists, teachers, test developers) and (2) what options are available for eliciting and representing knowledge in a field. To the first issue, two different approaches have been reported. one has the expert create a unique knowledge base relevant to a particular subject matter domain. These domains are usually quite narrow (such as particular microcircuitry) rather than similar to school subject matter (English literature). Thus, the question of extension of this approach to real school-based learning is at issue. Another possibility is the use of so-called expert tools. EMYCIN, (Heuristic Programming Project, Stanford), ROSIE (Rand Corporation), ART (Inference Corporation) and KEE (Intellicorp) are examples of systems designed to aid the efficient development the knowledge base without specifying subject matter (Richer, 1985). More recently, tools have been created for personal computer environments, e.g., M-1 (Teknowledge) and NEXPERT. These options may permit development of content for test and item generation. UCLA is currently exploring the feasibilityn of using tools of this sort to represent school subject matter (Baker & Freeman, 1987).

A second concern in AI related to assessment is representing the range of errors for diagnostic and instructional improvement purposes. Here, the work on Intelligent Computer-Assisted Instruction comes into play. ICAI depends upon the creation of a student model, a representation of the pattern of responses individual students make and a comparison of either their performance to expert problem solving strategies or a bug catalogue. The latter is a collection of incorrect procedures or "bugs," particularly as they apply to identifying micro errors or larger misconceptions (Johnson & Soloway, 1987). We believe this technology may be useful for the generation of wrong answer alternatives. Also relevant to this area is how test formats and psychometric quality get into such a system. Researchers at the Educational Testing Service (Freedle, 1985) have done some exploratory work on item generation using AI-based environments, presumed to be an improvement over non-AI assisted computer generation of test item formats.

We believe that the next five years will result in research which addresses overall how developments in ICAI can support the creation of test development systems. Such research will need to synthesize the science and application base, estimate the feasibility of building all or pieces of such a system, and to create small prototypes.

The AI Test Developer: a Developmental History

At UCLA, work began in 1985 on exploring the feasibility of an AI Test Developer (Baker, 1985). The original goal for the AI Developer was fairly grandiose. We were looking for a technology to decentralize testing - to pull some (but not all) of the responsibility of test design and publishing away from large, commercial entities and place sufficient testing expertise in the hands of the local educator. The benefits of such a system would be large. First, at least some fraction of school administered tests would be consistent with local views of curriculum and responsive to instructional experiences of students. Second, earlier research at UCLA (See, for example, Herman and Dorr-Breme, 1983; Baker, 1976) suggests that standardized test information is a relatively unused commodity in teachers' decision-making practices. However, teachers report that their

own tests provide the basis for data-driven instructional decisions. An AI Test Developer could provide the needed expertise and efficiency for teachers in the design of their own measures. Such a system would obviate the high cost training of teachers in test development (See Baker, 1978, Baker, et al, 1980; Rudman, et al, 1980.) Such a system should allow local teachers, district administrators and curriculum personnel, state managers, and private test developers to create tests that meet local curriculum needs. Such a global "expert" would fill-in deficient competencies of personnel, whether in item generation, quantitative analyses, or test interpretation. Of most interest are the two ICAI features mentioned eariler: the content domain issue and the assessment of student errors.

Critical components in the test developer. At the outset, the AI Developer was conceived as a complex, interacting system. However, a set of practical decisions modified the view. First, we decided to use commercially available expert system tools for the implementation of the developer. Secondly, we decided to constrain development hardware to likely user hardware in the short term (3 to 5 years) and limit ourselves to software compatible with personal computers in school districts and schools. Third, with a relatively scant set of resources, we decided to explore what expertise (other than the main test design function) was needed. Interviews with school district evaluation managers, personnel in private test development, and academic experts in achievement measurement provided an extensive list of discrete topics. Our focus then shifted from developing an integrated, memory eating monster to a set of test expert associates - the Test Expert Asssociate System (TEAS) During 1987, the first prototype of TEAS was undertaken with the expertise represented of Ronald Hambleton of the University of Massachussets. Using the M-1 expert tool, Hambleton dealt with the problem of the reliability of criterion-referenced tests .* Following the complete encoding of the rules gleaned from Hambleton, the system will be presented a set of problems to solve and its answers will be validated by

---

*Critical assistance in the activity has been provided by Dean Slawson and Zhonmin Li.

independent trials by Hambleton and two other psychometric experts. Then the system will be tested by school district personnel in order to document the utility of the format, the comprehensiveness of the advice, and their reaction to the system itself. At the same time, we carefully tracked time and cost of the design of the TEAS prototype to determine the feasibility of subsequent effort.

With a short lag, a second TEAS module is under development. Here it is the intent to attempt to represent a portion of school subject matter in order to determine whether it can be used as a generation context for test items. We have selected speeches from American History, particularly the Lincoln-Douglas Debates. We are interested in whether the original idea of the test developer (as an item generator) can be implemented in a low-cost environment. We are also interested in seeing whether we can find a way to use the TEAS component to help us generate criteria for adequate student essay responses, another critical measurement problem. The TEAS work is in process and will undoubtedly be affected by advances in software, predisposition to technology use, and research in cognitive science. An area of intense interest for us will be the future developments in natural language interfaces and understanding. To the extent the natural language field matures, testing may become less circumscribed, constrained, and formal and its development more distributed. We still feel we have the right goal (although, like ICAI designers, we view it as a context rather than a product to be engineered)—the development of a system that uses school subject matter knowledge bases, a system that could be standardized and shared. Assesssment devices would grow from these knowledge bases and might differ in symbolic representation presented or elicited from the learner and capitalize on student individual differences.

Conclusion

We have attempted to take a Janus view—of the ICAI field on the one hand and measurement and evaluation on the other. We have described how evaluation and measurement might be useful to the improvement of ICAI design and function and have

provided the few examples from our own work. We have also discussed new work in progress on the application of AI technology (TEAS) for the intermediate good of educational quality, as a resource to improve the measurement of achievement. Neither of these areas, either ICAI or AI-based measurement has a secure future. They may merely be side trips on a longer, more important educational journey. Of importance, however, is to analyze the processes involved in their development, and keep the good ideas. By taking both critical and empirical perspectives, we may be able to find productive, perhaps technological ways to our diverse educational goals.

REFERENCES

Alessi, S.M., & Trollip, S.R. (1985). *Computer-based instruction: Methods and development.* Englewood Cliffs, NJ: Prentice-Hall.

Anderson, R.J., Boyle, C.F., & Reiser, B.J. (1985). Intelligent tutoring systems. *Science, 228,* 456-462.

Baker, E.L. (1974). Formative evaluation in instruction. In J. Popham (Ed.), *Evaluation in education..* Berkeley, CA: McCutchan.

Baker, E.L. (1976). *The evaluation of the California Early Childhood Education Program* (Vol. I), 273 pp. Los Angeles: UCLA Center for the Study of Evaluation.

Baker, E.L. (1978). The evaluation and research of multi-project programs: Program component analysis. *Studies in Educational Evaluation.*

Baker, E.L., & Alkin, M.C. (1973). Formative evaluation in instructional development. *AV Communication Review, 21*(4).

Baker, E.L., Bradley, C., Aschbacher, P., & Feifer, R. (1985). *Intelligent computer-assisted instruction (ICAI) study.* Final Report to Jet Propulsion Laboratory. Los Angeles: UCLA Center for the Study of Evaluation.

Baker, E.L., & Herman, J.L. (1983). Task structure design: Beyond linkage. *Journal of Educational measurement, 20*(2), 149-164.

Baker, E.L., & Herman, J.L. (1985). Educational evaluation: Emergent needs for research. *Evaluation Comment, 7*(2).

Baker, E.L., & Linn, R.L. (1986, April). *New testing technologies.* Sherman Oaks, CA: Advande Design Information, Inc.

Baker, E.L., & O'Neil, H.F., Jr. (1987). Assessing instructional outcomes. In R.M. Gagné (Ed.), *Instructional Technology: Foundations.* Hillsdale, NJ: Lawrence Erlbaum.

Baker, E.L., Polin, L., & Burry, J. (1980). *Making, choosing, and using tests: A practiicuum on domain-referenced tests.* Report to the National Institute of Education. Los Angeles: UCLA Center for the Study of Evaluation.

Baker. E.L., & Quellmalz, E. (1980). *Educational testing and evaluation: Design, analysis and policy.* Beverly Hills, CA: Sage.

Baker, E.L., & Saloutos, W.A. (1974). *Formative evaluation of instruction.* Los Angeles: UCLA Center for the Study of Evaluation.

Barr, A., & Feigenbaum, E.A. (Eds.) (1982). *The handbook of artificial intelligence* (Vol. 2). Los Altos, CA: William Kaufmann.

Bloom, B.S. (1984 June/July). The 2 Sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher, 13*(6).

Bobrow, D.G., & Stefik, M. (1983). *The LOOPS manual.* Palo Alto: Xerox.

Bright, J.R. (1964). *Research, development, and technical innovation—an introduction.* Homewood, IL: Richard D. Irwin.

Brown, J.S., & Burton, R.R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science, 2*, 155-192.

Brown, J.S., Burton, R.R., & de Kleer, J. (1982). Knowledge engineering and pedagogical techniques in Sophie I, II, and III. In D. Sleeman & J.S. Brown (Eds.), *Intelligent tutoring systems.* New York: Academic Press.

Buchanan, B.G. (1981). *Rsearch on expert systems.* Report number CS-81-837, Computer Science Department, Stanford University.

Burstein, L., Baker, E.L., Aschbacher, P., & Keesling, J.K. (1985). *Using state test data for national indicators of educational quality: A feasibility study.* Los Angeles: UCLA Center for the Study of Evaluation.

Burton, R.R., & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman and J.S. Brown (Eds.), *Intelligent tutoring systems.* New York: Academic Press.

Carbonell, J.R., & Collins, A. (1973). Natural semantics in artificial intelligence. *International Jountal of Computer Aided Instruction, 3*, 344-352.

Carr, B., & Goldstein, I.P. (1977). *Overlays: A theory of modeling for computer aided instruction.* (Artificial Intelligence Memo 406). Cambridge, MA: MIT.

Carroll, J.M., & Rosson, M.B. (1984). *Usability specifications as a tool in interative development.* (Research Report RC 10437 [#46642] 4/3/84). Yorktown Heights, NY: IBM Watson Research Center, Computer Science Department.

Chi, M.T.H., Glaser, R., & Rees, E. (1982). Expertise in problem solving. In R.j. Sternberg (Ed.), *Advances in the psychology of human intelligence* (Vol. 1, pp. 7-76). Hillsddale, NJ: Erlbaum.

Chi, M.T.H., Bassok, M., Lewis, M.W., Reimann, P., & Glaser, R. (1987 November). *Self-explanations: How students study and use examples in learning to solve problems* (Technical Report No. 9). Pittsburgh, PA: University of Pittsburgh, Learning Research and Development Center.

Clancey, W.J. (1979). Tutorial rules for guiding a case method dialogue. *International journal of Man-Machine Studies, 11*, 25-50.

Clancey, W.J. (1979). *Transfer of rule-based expertise through tutorial dialogue.* Report No. CS-769, Computer Science Department, Stanford University.

Clancy, W.J. (1981). *The epistemology of a rule-based expert system: A framework for explanation.* Report number CA 81-896, Computer Science Department, Stanford University.

Clancy, W.J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman and J.S. Brown (Eds.), *Intelligent tutoring systems.* London: Academic Press.

Clement, J., Lockhead, J., & Soloway, E. (1980). *Positive effects of computer programming on students' understanding of variables and equations.* Proceedings of the National Association for Computing Machinery, Nashville.

Corno, L., & Snow, R.E. (1986). Adapting teaching to individual differences among learners. In M.C. Wittrock (Ed.), *Handbook of research on teaching* (3rd ed.). New York: Macmillan.

Duda, R.O., & Gaschnig, J.G. (1981). Knowledge-based expert systems come of age. *BYTE, 6,* 238-281.

Ellis, J., Wulfeck, W.H., & Fredericks, P.S. (1979). *The instructional quality inventory: II. User's manual.* (NPRDC SR 79-24). San Diego, CA: Navy Personnel Research and Developmen Center. (AD-A083-678).

Fletcher, J.D. (1985). Intelligent instructional systems in training. IN S. A. Andriole (Ed.), *Applications in artificial intelligence.* Princeton, NJ: Petrocelli.

Floden, R.E., Freeman, D.J., Porter, A.C., & Schmidt, W.H. (1980). Don't they all measure the same thing? Consequences of selecting standardized tests. IN E.L. Baker & E. Quellmalz (Eds.), *Design analysis and policy in testing and evaluation.* Beverly Hills, CA: Sage.

Freedle, R. (1985). *Implications of language programs in artificial intelligence for testing issues.* Final Report, Project 599-63. Princeton, NJ: Educational Testing Service.

Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E., & Terry, A. (1983). Evaluation of expert systems: Issues and case studies. In F. Hayes-Roth, D.A. Waterman, & D.B. Lenat (Eds.), *Building expert systems.* Reading, MA: Addison-Wesley.

Gamble, A., & Page, C.V. (1980). *IJ Man-Machine Studies, 12,* 259-282.

Glennan, T.K., Jr. (1967). Issues in the choice of development policies. In T. Manschak, T.K. Glennan, Jr., & R. Summers (Eds.), *Strategies for research and development.* New York: Springer-Verlag.

Hayes-Roth, B. (1980). *Human planning processes.* Report number R-2670, RAND Corporation, Santa Monica, CA.

Hedl, J.J., Jr., & O'Neil, H.F., Jr. (1977). Reduction of state anxiety via instructional design in computer-based learning environments. In J. Seiber, H.F. O'Neil, Jr., & S. Tobias, (Eds.), *Anxiety, learning and instruction.* New York: Lea/Wiley.

Herman, J., & Cabello, B. (1983). *An analysis of the match between the California Assessment Program and commonly used standardized tests.* Los Angeles: UCLA Center for the Study of Evaluation.

Herman, J., & Dorr-Bremme, D. (1983). Uses of testing in the schools: A national profile. *New Directions for Testing and Measurement.* (No. 19). San Francisco: Jossey-Bass.

Hively, W., Patterson, J., & Page, S. (1968). A "universe defined" system of arithmetic achievement testing. *Journal of Educational measurement, 5*(4), 275-290.

Hollan, J.D., Hutchins, E.L., & Weitzman, L. (1984). STEAMER: An interactive inspectable simulation-based training system. *Artificial Intelligence magazine, 5*, 15-27.

Johnson, W.L., & Soloway, E. (1983). *PROUST: Knowledge-based program understanding.* Report No. 285, Computer Science Depatment, Yale University.

Johnson, W.L., & Soloway, E. (1987) PROUST: An automatic debugger for PASCAL programs. In G.P. Kearsley (Ed.), *Artificial intelligence: Applications and methodology.* Reading, MA: Addison-Wesley.

Kieras, D.E., & Bovair, S. (1983). *The role of a mental model in learning to operate a device.* (Technical Report No. 13 RZ/DP/TR-83/ONR-13). University of Arizona, Department of Psychology.

Lea, W. (Ed.). (1980). *Trends in speech recognition.* Englewood Cliffs, NJ: Prentice-Hall.

Markle, S.M. (1967). Empirical testing of programs. In P.C. Lange (Ed.), *Programmed instruction.* Sixty-sixth Yearbook of the National Society for the Study of Education, Part II. Chicago: University of Chicago Press.

McArthur, D., Stasz, C., Hotta, J. (1987). Learning problem-solving skills in algebra. *Journal of Educational Technology Systems, 15*(3), 303-324.

Merrill, M.D., & Tennyson, R.D. (1977). *Teaching concepts: An instructional design guide*. Englewood Cliffs, NJ: Educational Technology Publications..

Merrill, P.F., Tolman, M.N., Christensen, L., Hammons, K., Vincent, B.R., & Reynolds, P.L. (1986). *Computers in educatino*. Englewood Cliffs, NJ: Prentice-Hall.

O'Neil, H.F., Jr. (Ed.). (1979). *Procedures for instructional systems development* . New York: Academic Press.

O'Neil, H.F., Jr., Anderson, C.L., & Freeman, J.A. (1986). Research in teaching in the Armed Forces. In M.C. Wittrock (Ed.), *Handbook of research on teaching* (3rd ed.). New York: Macmillan.

O'Neil, H.F., Jr., & Paris, J. (1981). Introduction and overview of computer-based instruction. In H.F. O'Neil, Jr. (Ed.), *Computer-based instruction*. New York: Academic Press.

O'Neil, H.F., Jr., Slawson, D.A., & Baker, E.L. (1987). *First application's domain-independent and domain-specific instructional strategies for knowledge bases (August 31 1987)*. Sherman Oaks, CA: Advance Design Information, Inc.

O'Neil, H.F., Jr., & Spielberger, C.D. (Eds.). (1979). *Cognitive and affective learning strategies*. New York: Academic Press.

Park, P., Perez, R.S., & Seidel, R.J. (1987). Intelligent CAI: Old wine in new bottles or a new vintage? In G.P. Kearsley (Ed.), *Artificial intelligence: Applications and methodology*. Reading, MA: Addison-Wesley.

Reigeluth, C.M. (1983). Instructional design: What is it and why is it? In C.M. Reigeluth, (Ed.), *Instructional-design theories and models: An overview of their current status*. Hillsdale, NJ: Lawrence Erlbaum.

Reigeluth, C.M. (Ed.) (1987). *Instructional theories in action: Lessons illustrating selected theories and models*. Hillsdale, NJ: Lawrence Erlbaum.

Richer, M.H. (1985). *Evaluating the existing tools for developing knowledge-based systems*. Knowledge Systems Laboratory Report No. KSL 85-19. Stanford, CA: Stanford University, Stanford Knowledge Systems Laboratory.

Rudman, et al. (1980). *Integrating assessment with instruction: A review 1922-1980*. Research Series, No. 75. East Lansing, MI: Institute for Research on Teaching.

Schank, R.C. (1980). Language and memory. *Cognitive Science, 4*, 243-284.

Scriven, M. (1967). The methodology of evaluation. In R.W. Tyler, R.M. Gagné, & M. Scriven (Eds.), *Perspectives of curriculum evaluation. AERA Monograph Series on Curriculum Evaluation, No. 1*. Chicago: Rand McNally.

Sternberg, R. (Ed.). (1982). *Advances in the psychology of human intelligence*. Hillsdale, NJ: Lawrence Erlbaum.

Stevens, A.L., Collins, A., & Goldin, S. (1978). *Diagnosing students' misconceptions in causal models*. Report No. 3786, Cambridge, MA, Bolt, Beranek, & Newman.

Stevens, A., & Steinberg, C. (1981). *Project STEAMER*, NPRDC Technical Note Number 82-21. San Diego, CA: Navy Personnel Research and Development Center.

Weinstein, C.F., & Mayer, R.F. (1986). The teachning of learning strategies. In M.C. Wittrock (Ed.), *Handbook of research on teaching* (3rd ed.). New York: Macmillan.

Winston, P.H. (Ed.). (1975). *The psychol;ogy of computer vision*. New York: McGraw-Hill.

Winston, P.H. (1977). *Artificial intelligence*. Reading, MA: Addison-Wesley.

Woods, W.A. (1983). What's important about knowledge representation? *Computer, 16*, 22-29.

Figure 1

DESIRABLE PROPERTIES OF A HUMAN TUTOR

- The tutor causes the problem solving heuristics of the student to converge to those of the tutor.
- The tutor chooses appropriate examples and problems for the student.
- The tutor can work arbitrary examples chosen by the student.
- The tutor is able to adjust to different student backgrounds.
- The tutor is able to measure the student's progress.
- The tutor can review previously learned material with the student as the need arises.

Adapted from Gamble & Page (1980)

Figure 2

FORMATIVE EVALUATION ACTIVITY

1. Check ICAI design again its specifications.
2. Check validity of instructional strategies in tutor with research literature.
3. Conduct feasibility review with instructor
4. Conduct feasibility test with student(s)
    - one-on-one testing
    - small group testing
4. Assess instructional effectiveness.
    - cognitive
    - affective
5. Assess unanticipated outcomes.
6. Conduct revision.

FIGURE 3

ICAI METAPHORS

SETTING	STUDENT ROLE	EVALUATION FOCUS
Laboratory	Applied scientist	Problem solving ability increased
Classroom	Learner	Learning increased
Arcade	Game player	Enjoyment and learning increased
Workbench	Troubleshooter	Ability to fix faults increased
Expert System or Automated Job Performance Aid	Human System Component	System goal achieved

TABLE 1

EVALUATION QUESTIONS

I. Are the measures and procedures planned and used for formative and summative evaluation providing a fair test of the ICAI system?

II. Does the ICAI system meet its multiple goals?

a. Generalization

1. Does the prototype provide the desired level of education/training?

2. Is this level maintained or improved as the prototype addresses more complex education/training missions; greater numbers of students; distributed sites?

3. Will the prototype easily generalize (or adapt) to other content areas (e.g., algebra to English)?

b. Technology Push

1. Does the development of the existing hardware/software components for the system (e.g., knowledge representation, graphics) contribute to the capability for future education/training?

2. Have other technological approaches to education/training (e.g., metacognitive skill training) been considered and integrated into planned future prototype?

c. Unplanned Outcomes (Side-effects Analysis)

1. Does the system create requirements to train teachers for new role (e.g., expert remediator)?

2. Will intensive data collection systems permit answers to "old" questions, e.g., relative value of discovery learning, estimation of transfer both near and far?

3. Is the prototype a good environment to validate analytic techniques to predict the education/training effectiveness?

4. Will intensive data collection permit answers to "new" questions from cognitive science (e.g., analysis of misconceptions or bugs; differences between experts and novices; role of mental models in proficiency)?

TABLE 2

CAI LESSON EVALUATION TECHNIQUES

QUALITY REVIEW

Check the language and grammar [e.g., appropriate reading level].
Check the surface features [e.g., uncluttered displays].
Check questions and menus [e.g., making a choice is clear].
Check all invisible functions [e.g. appropriate student records kept].
Check the subject matter content [e.g., information is accurate].
Check the off-line material [e.g., directions in operator manual are clear].
Revise the lesson.
Apply the same quality-review procedure to all revisions.

PILOT TESTING

Enlist about three helpers [i.e., representative of potential students].
Explain pilot-testing procedures [e.g., encourage note-taking].
Find out how much they know about the subject matter.
Observe them go through the lesson.
Interview them afterwards.
Revise the lesson.
Pilot-test all revised lessons.

VALIDATION

Use the lesson in the setting for which it was designed.
Use the lesson with students for which it was designed.
Evaluate how the students perform in the setting for which you are preparing them.
Obtain as much performance data as you can from different sources.
Obtain data on student achievement attributable to the lesson.
Obtain data on student attitudes toward the lesson.

Adapted from Alessi & Trollip (1985), p. 393.

TABLE 3

AI FEATURES IN ICAI SYSTEMS

TOPIC	EXAMPLES
Knowledge representation techniques	Production rules, frames, networks
Reasoning mechanisms	Backward and forward chaining, inheritance
Development environment	User-interface, editors and debuggers, documentation and on-line help systems
Rapid prototypes	Rapidly developed simulation, exhibit functionality, convey requirements; not meant to be operational systems
Student modeling methods	Overlay, buggy, individual differences
Knowledge acquisition techniques	"Shells," knowledge-base elicitors
Validation tools	check integrity of knowledge base to identify conflicting rules or syntactical errors
Cost factors	Price of software, support, training, required hardware, skilled personnel
Expert tutor	Domain-independent instructional strategies
Cognitive or process model	Model of how system accomplishes its tasks, may be based on models of human reasoning (e.g., schema)
Languages	LISP, PROLOG